10-3 關聯式資料庫與SQLite

- 10-3-1 認識關聯式資料庫
- 10-3-2 SQLite資料庫引擎
- 10-3-3 SQL語言的基礎

10-3 關聯式資料庫與SQLite

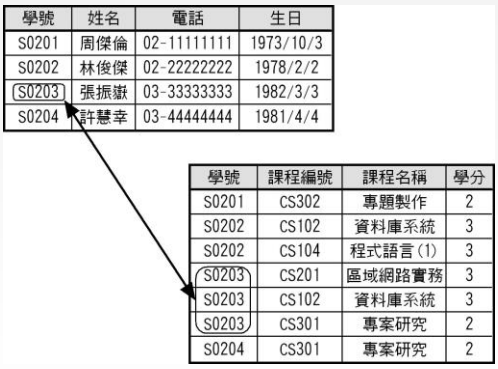
- 資料庫（Database）是一種資料儲存單位，一些經過組織的資料集合，眾多出勤管理系統、倉庫管理系統、進銷存系統或小至錄影帶店管理系統，這些應用程式都屬於不同應用的資料庫系統。
- 資料庫系統是由資料庫和資料庫管理系統所組成，資料庫管理系統是一套管理資料庫的應用程式。

10-3-1 認識關聯式資料庫 – 說明

- 關聯式資料庫系統（Relational Database System）是目前資料庫系統的主流，市面上大部分資料庫管理系統都屬於關聯式資料庫管理系統（Relational Database Management System），例如：Access、MySQL、SQL Server和Oracle等。
- 關聯式資料庫（Relational Database）是由一個或多個資料表所組成，在多個資料表間使用欄位的資料值來建立連接，以便實作資料表之間的關聯性。

10-3-1 認識關聯式資料庫 – 圖例

- 在關聯式資料庫是使用二維表格的資料表來儲存記錄資料，在各資料表間使用欄位值建立關聯性，透過關聯性來存取其他資料表的資料。例如：使用【學號】欄位值建立兩個資料表之間的關聯性，如右圖所示：



學號	姓名	電話	生日
S0201	周傑倫	02-11111111	1973/10/3
S0202	林俊傑	02-22222222	1978/2/2
S0203	張振嶽	03-33333333	1982/3/3
S0204	許慧幸	03-44444444	1981/4/4

學號	課程編號	課程名稱	學分
S0201	CS302	專題製作	2
S0202	CS102	資料庫系統	3
S0202	CS104	程式語言(1)	3
S0203	CS201	區域網路實務	3
S0203	CS102	資料庫系統	3
S0203	CS301	專案研究	2
S0204	CS301	專案研究	2

10-3-1 認識關聯式資料庫 – 組成

- 關聯式資料庫的資料是儲存在資料庫的「資料表」（Tables），每一個資料表使用「欄位」（Fields）分類成多個群組，每一個群組是一筆「記錄」（Records），例如：通訊錄資料表的記錄，如下表所示：

編號	姓名	地址	電話	生日	電子郵件地址
1	陳會安	新北市五股區成泰路1000號	02-11111111	1967/9/3	hueyan@ms2.hinet.net
2	江小魚	新北市中和區景平路1000號	02-22222222	1978/2/2	jane@ms1.hinet.net
3	劉得華	桃園市三民路1000號	02-33333333	1982/3/3	lu@tpts2.seed.net.te
4	郭富成	台中市中港路三段500號	03-44444444	1981/4/4	ko@gcn.net.tw
5	離明	台南市中正路1000號	04-55555555	1978/5/5	light@ms11.hinet.net
6	張學有	高雄市四維路1000號	05-66666666	1979/6/6	geo@ms10.hinet.net
7	陳大安	台北市羅斯福路1000號	02-99999999	1979/9/9	an@gcn.net.tw

10-3-2 SQLite資料庫引擎

- SQLite是目前世界上最廣泛使用的免費資料庫引擎，一套實作大部分SQL 92標準的函數庫，它不需要管理、不需要伺服器、也不需要安裝設定，不但體積輕巧，而且還是一套支援交易（Transaction）的SQL資料庫引擎，其官方網址為：http://www.sqlite.org/。
- SQLite是Android作業系統內建的資料庫系統，其主要特點如下所示：
  - SQLite資料庫只是一個檔案，可以直接使用檔案權限來管理資料庫，而不用自行處理資料庫的使用者權限管理，所以沒有提供SQL語言的DCL存取控制。
  - 單一檔案的SQLite資料庫，可以讓Android應用程式很容易進行安裝，而且不用特別進行資料庫系統的設定與管理。
  - SQLite不需要啟動，換句話說，不會浪費行動裝置的記憶體資源。

10-3-3 SQL語言的基礎 – 說明

- 「SQL」（Structured Query Language）為「ANSI」（American National Standards Institute）標準的資料庫語言，可以存取和更新資料庫的記錄資料。SQLite支援的SQL指令主要分成兩大類，如下所示：
  - 資料定義語言（Data Definition Language，DDL）：建立資料表和定義資料表欄位。
  - 資料操作語言（Data Manipulation Language，DML）：資料表記錄的查詢、插入、刪除和更新。

10-3-3 SQL語言的基礎 – CREATE TABLE建立資料表

- CREATE TABLE 指令可以在資料庫建立資料表，如下所示：

```
CREATE TABLE titles (
  _id integer primary key autoincrement,
  title text no null,
  price real no null
)
```

10-3-3 SQL語言的基礎 – SELECT查詢記錄(語法)

- 在SQL語言的資料查詢指令只有一個SELECT指令，其基本語法如下所示：

  SELECT 欄位1, 欄位2 FROM 資料表

  WHERE conditions

  ORDER BY 欄位清單

- SELECT指令的欄位1~2為記錄的欄位，conditions為查詢條件，使用口語來說就是「從資料表取回符合WHERE子句條件的記錄，顯示欄位1和2，並且以ORDER BY子句的欄位來排序」。

10-3-3 SQL語言的基礎 – SELECT查詢記錄("*"符號)

- SELECT指令可以使用"*"符號代表資料表的所有欄位，表示取回資料表記錄的所有欄位，如下所示：

  SELECT * FROM titles

10-3-3 SQL語言的基礎 – SELECT查詢記錄(WHERE子句-1)

- SQL查詢如果是單一條件，在WHERE子句條件的基本規則和範例，如下所示：
  - 文字欄位需要使用單引號括起，例如：書名為Java時的SQL指令字串，如下所示：

```
SELECT * FROM titles WHERE title='Java'
```

  - 數字欄位不需要使用單引號括起，例如：價格為600元的SQL指令，如下所示：

```
SELECT * FROM titles WHERE price=600
```

10-3-3 SQL語言的基礎 – SELECT查詢記錄(WHERE子句-2)

- 文字和備註欄位可以使用【LIKE】包含運算子，只需包含此字串即符合條件，再配合"%"或"_"萬用字元，可以代表任何字串或單一字元，只需包含的子字串就符合條件。例如：查詢擁有Java子字串圖書資料的SQL指令，如下所示：

```
SELECT * FROM titles WHERE titles LIKE '%Java%'
```

- 數字或日期/時間欄位可以使用<>、>、<、>=和<=不等於、大於、小於、大於等於和小於等於等運算子建立多樣化的查詢條件。

10-3-3 SQL語言的基礎 – INSERT新增記錄

- SQL新增記錄INSERT指令可以新增一筆記錄例如：在titles資料表新增一筆記錄的SQL指令，如下所示：

```
INSERT INTO titles (title, price)
VALUES ('Access', 600)
```

- SQL指令的括號中是欄位清單（不用全部，但需包含所有not null欄位），字串欄位值使用單引號括起，數字沒有。

10-3-3 SQL語言的基礎 – UPDATE更新記錄

- SQL更新記錄UPDATE指令是將資料表內符合條件的記錄，更新其欄位內容，例如：在titles資料表更改書價的SQL指令，如下所示：

```
UPDATE titles SET price =650
WHERE title='Access'
```

- SQL指令的WHERE條件為書名title欄位，然後使用SET子句更新price欄位資料。

10-3-3 SQL語言的基礎 – DELETE刪除記錄

- SQL刪除記錄DELETE指令是將資料表內符合條件的記錄刪除掉。例如：在titles資料表刪除記錄的SQL指令，如下所示：

  DELETE FROM titles WHERE title='Access'

- SQL指令的WHERE條件為書名title欄位，也就是將符合書名條件的圖書記錄刪除掉。

10-4 SQLite資料庫的使用

- 10-4-1 使用SQLiteOpenHelper類別建立資料庫與資料表
- 10-4-2 使用SQLiteDatabase類別存取資料表的記錄資料
- 10-4-3 使用SQL指令存取資料庫

10-4-1 使用SQLiteOpenHelper類別建立資料庫與資料表 – 說明

- SQLiteOpenHelper類別是一個幫助我們存取SQLite資料庫的幫助類別（Helper Class），Android應用程式建立SQLite資料庫就是繼承SQLiteOpenHelper類別來覆寫相關方法，事實上，其主要目的是讓我們可以在SQLite資料庫新增資料表（因為資料庫的建立已經在父類別實作）。

10-4-1 使用SQLiteOpenHelper類別建立資料庫與資料表 - 繼承SQLiteOpenHelper類別1

- 在MyDBHelper（自行命名）子類別需要新增建構子，覆寫onCreate()和onUpgrade()方法，首先是建構子，如下所示：

```
public class MyDBHelper extends SQLiteOpenHelper {
  private static final String DATABASE_NAME = "MyBooks";
  private static final int DATABASE_VERSION = 1;
  public MyDBHelper(Context context) {
    super(context, DATABASE_NAME,
                null, DATABASE_VERSION);
  }
```

10-4-1 使用SQLiteOpenHelper類別建立資料庫與資料表 - 繼承SQLiteOpenHelper類別2

- 我們需要覆寫onCreate()和onUpgrade()方法，如下所示：

```
@Override
public void onCreate(SQLiteDatabase db) {
  db.execSQL("CREATE TABLE titles (_id integer " +
      "primary key autoincrement, " +
      "title text no null, price real no null)");
}
@Override
public void onUpgrade(SQLiteDatabase db,
            int oldVersion, int newVersion) {
  db.execSQL("DROP TABLE IF EXISTS titles");
  onCreate(db);
}
}
```

10-4-1 使用SQLiteOpenHelper類別建立資料庫與資料表 - 相關方法

- SQLiteOpenHelper類別提供相關方法來開啟和關閉資料庫，如下表所示：

方法	說明
getReadableDatabase()	建立或開啟（如果存在）一個唯讀資料庫，成功開啟傳回SQLiteDatabase物件
getWritableDatabase()	建立或開啟（如果存在）一個讀寫資料庫，成功開啟傳回SQLiteDatabase物件
close()	關閉開啟的資料庫

10-4-2 使用SQLiteDatabase類別存取資料表的記錄資料 – 說明

- SQLiteDatabase類別提供相關方法來新增、更新和刪除資料表的記錄資料。
- 為了完整說明SQLite資料庫的使用，從開啟和關閉資料庫開始，一一說明如何存取資料庫的記錄資料。

10-4-2 使用SQLiteDatabase類別存取資料表的記錄資料 - 開啟可讀寫的資料庫

- 因為需要在資料庫新增、更新和刪除記錄，所以開啟可讀寫資料庫，通常是在Activity類別的onCreate()方法開啟資料庫，如下所示：

```
dbHelper = new MyDBHelper(this);
db = dbHelper.getWritableDatabase();
```

- 程式碼在建立MyDBHelper物件後，呼叫getWritableDatabase()方法來取得SQLiteDatabase物件的資料庫。

10-4-2 使用SQLiteDatabase類別存取資料表的記錄資料 - 關閉資料庫

- 關閉資料庫通常是在onStop()方法，只需呼叫SQLiteDatabase類別的close()方法，就可以關閉資料庫，如下所示：

```
db.close();
```

10-4-2 使用SQLiteDatabase類別存取資料表的記錄資料 - 新增記錄

- SQLiteDatabase物件可以使用insert()方法來新增記錄，首先，我們需要使用ContentValues類別建立欄位值，使用put()方法加入欄位，如下所示：

```
long id;
ContentValues cv = new ContentValues();
cv.put("title", txtTitle.getText().toString());
double price =
   Double.parseDouble(txtPrice.getText().toString());
cv.put("price", price);
```

- 在建立後，呼叫insert()方法來新增記錄，如下所示：

```
id = db.insert(DATABASE_TABLE, null, cv);
```

10-4-2 使用SQLiteDatabase類別存取資料表的記錄資料 - 更新記錄

- SQLiteDatabase物件可以使用update()方法來更新記錄，同樣需要使用ContentValues類別建立更新的欄位值，如下所示：

```
ContentValues cv = new ContentValues();
double price = Double.parseDouble(
    txtNewPrice.getText().toString());
cv.put("price", price);
count = db.update(DATABASE_TABLE, cv,
        "title='" + title + "'", null);
```

10-4-2 使用SQLiteDatabase類別存取資料表的記錄資料 - 刪除記錄

- SQLiteDatabase物件可以使用delete()方法來刪除記錄，如下所示：

```
count = db.delete(DATABASE_TABLE,
        "title='" + title + "'", null);
```

- 方法的傳回值是影響的記錄數，第1個參數是資料表名稱，第2個就是WHERE子句的刪除條件，如果是參數條件字串，其參數值就是最後一個參數。

10-4-2 使用SQLiteDatabase類別存取資料表的記錄資料 – 查詢記錄1

- 一般來說，如果是取出單筆的記錄資料，我們可以在query()方法加上WHERE子句的條件（不含WHERE本身），如下所示：

```
Cursor c = db.query(DATABASE_TABLE, colNames,
        "_id = " + rowId , null, null, null,null);
```

- 上述程式碼指定第3個參數的條件為_id欄位值等於rowId變數，可以取回指定記錄編號的記錄資料。

10-4-2 使用SQLiteDatabase類別存取資料表的記錄資料 – 查詢記錄2

- 在取得Cursor物件後，我們就可以移動記錄指標來取得查詢結果的每一筆記錄，如下所示：

```
c.moveToFirst();
for (int i = 0; i < c.getCount(); i++) {
  str += c.getString(
   c.getColumnIndex(colNames[0])) + "\t\t";
  str += c.getString(1) + "\t\t";
  str += c.getString(2) + "\n";
  c.moveToNext();
}
```

- 程式碼首先呼叫moveToFirst()方法移至第1筆記錄，getCount()方法傳回記錄數，然後使用for迴圈來走訪每一筆記錄，moveToNext()方法可以移至下一筆。

10-4-3 使用SQL指令存取資料庫－execSQL()方法

- 在第10-4-2節是使用SQLiteDatabase物件的方法來新增、更新、刪除和查詢記錄資料，事實上，我們也可以如同第10-4-1節建立資料表一般，直接使用execSQL()方法下達SQL指令來存取資料庫，如下所示：

```
db.execSQL("INSERT INTO " + DATABASE_TABLE + " (" +
   "title, price) VALUES ('" + txtTitle.getText().toString() +
   "'," + txtPrice.getText().toString() + ")");
```

- 上述程式碼建立SQL指令的INSERT指令來新增記錄，同理，可以建立UPDATE指令來更新記錄；DELETE指令刪除記錄資料。

10-4-3 使用SQL指令存取資料庫－rawQuery()方法

- 查詢部分是使用rawQuery()方法下達SELECT指令，如下所示：

```
Cursor c = db.rawQuery("SELECT * FROM " +
                  DATABASE_TABLE, null);
colNames = c.getColumnNames();
for (int i = 0; i < colNames.length; i++)
  str += colNames[i] + "\t\t";
str += "\n";
```

- rawQuery()方法的第1個參數是SELECT指令，第2個參數是當第1個參數是參數查詢時，指定「?」符號取代的參數值，可以傳回Cursor物件，此時可以使用getColumnNames()方法取得欄位名稱陣列，然後使用for迴圈顯示查詢結果。